

Flash is a permissionless protocol allowing everyone to stake **\$FLASH** and earn instant upfront yield. Because of its permissionless nature, it will exist for as long as Ethereum does.

Features:

- Stake **\$FLASH** and redirect the yield to EOA or a contract
- A percentage of the generated yield is matched and put in an address. The match ratio can be as low as **0%** and as high as **20%**
- Unstake **\$FLASH** after the expire period is over
- Unstake **\$FLASH** before the expire period is over, but burn percentage of the staked amount based on the remaining time

- The **FPY** (Flash Percentage Yield) is calculated with the formula:

$$\left(1 - \frac{\text{total_staked_amount} + \text{new_staked_amount}}{\text{total_supply}}\right) \div 2$$

- The instant yield reward is calculated with the formula:

$$\frac{\text{new_staked_amount} \times \text{expiration_in_seconds} \times \text{FPY}}{86400 \times 365}$$

- The burn amount when unstaking early is calculated with the formula:

$$\frac{\text{staked_amount} \times \text{remaining_time_in_seconds}}{\text{total_time_in_seconds}} \times \frac{\text{total_staked_amount} - \text{staked_amount}}{\text{total_supply}}$$

How it works

Flash is made up of a single solidity contract. The purpose is to allow staking and earning instant upfront yield that can be redirected to an EOA or a contract. If the yield receiver is a contract, it should provide the interface function `receiveFlash(bytes32, uint256, uint256, uint256, address, bytes)` external returns (uint256). The flash protocol does not care what happens after calling `receiveFlash`

